Applied Mathematics and Computation 252 (2015) 155–165

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

A review of chaos-based firefly algorithms: Perspectives and research challenges

Iztok Fister Jr. [a], Matjaž Perc [b,c,d,*], Salahuddin M. Kamal [c], Iztok Fister [a]

[a] Faculty of Electrical Engineering and Computer Science, University of Maribor, Smetanova 17, SI-2000 Maribor, Slovenia
[b] Faculty of Natural Sciences and Mathematics, University of Maribor, Koroška cesta 160, SI-2000 Maribor, Slovenia
[c] Department of Physics, Faculty of Science, King Abdulaziz University, Jeddah, Saudi Arabia
[d] CAMTP – Center for Applied Mathematics and Theoretical Physics, University of Maribor, Krekova 2, SI-2000 Maribor, Slovenia

ARTICLE INFO

Keywords:
Firefly algorithm
Chaos
Chaotic map
Optimization
Swarm intelligence

ABSTRACT

The firefly algorithm is a member of the swarm intelligence family of algorithms, which have recently showed impressive performances in solving optimization problems. The firefly algorithm, in particular, is applied for solving continuous and discrete optimization problems. In order to tackle different optimization problems efficiently and fast, many variants of the firefly algorithm have recently been developed. Very promising firefly versions use also chaotic maps in order to improve the randomness when generating new solutions and thereby increasing the diversity of the population. The aim of this review is to present a concise but comprehensive overview of firefly algorithms that are enhanced with chaotic maps, to describe in detail the advantages and pitfalls of the many different chaotic maps, as well as to outline promising avenues and open problems for future research.

1. Introduction

In the past, facing the real-world optimization problems was in the domain of mathematicians and engineers. They developed many mathematical methods for solving the optimization problems. The first algorithms solved the problems exactly by enumerating all the possible solutions, but rapidly algorithms for approximate (heuristic) solving of these problem have been emerged because of a huge time complexities of the exact methods. One of the more promising algorithms today are swarm intelligence (SI) based algorithms inspired with a collective behavior of some simple unintelligent insects or animals, who work together in order to be capable of solving the complex problems. For instance, a foraging of insects connects social living bees, ants and termites. In nature, one individual cannot survive, but when living together in colonies, individuals are stronger and thus capable of performing very complex tasks (e.g., huge mounds by termites). These colonies of insects act as decentralized, and self-organized systems that prevents a single insect to act alone.

A firefly algorithm (FA) is one of the younger member of SI-based algorithms that was introduced by Yang in [40] at 2008. Since its introduction, many researchers began working with FA. At the beginning, some modified variants were proposed that were applied for solving the continuous optimization [43], multimodal [41], constrained optimization [30], and later also for real-world problems [47,36,16,25]. Competition of FA with other well-known meta-heuristics led to the development of more robust and sophisticated FA variants. For example, Fister et al. in [12] proposed a memetic self-adaptive FA

* Corresponding author at: Faculty of Natural Sciences and Mathematics, University of Maribor, Koroška cesta 160, SI-2000 Maribor, Slovenia.
E-mail address: matjaz.perc@um.si (M. Perc).

http://dx.doi.org/10.1016/j.amc.2014.12.006
0096-3003/

(MFA), where values of control parameters are changed during the run. Their version of the FA was developed for combinatorial optimization, while Galvez and Iglesias [17] created a new memetic self-adaptive FA for the continuous optimization. Simultaneously, also a bunch of applications using FA were created [18,19,3].

Stochastic optimization algorithms search for optimal solutions by involving randomness in some constructive way [24]. In contrast, if optimization methods provide the same results when doing the same things, these methods are said to be deterministic [11]. If the deterministic system behaves unpredictably, it arrives at a phenomenon of chaos [11]. Consequently, randomness in SI algorithms plays a huge role because this phenomenon affects the exploration and exploitation in search process [5]. These companions of stochastic global search represent the two cornerstones of problem solving, i.e., exploration refers to moves for discovering entirely new regions of a search space, while exploitation refers to moves that focus searching the vicinity of promising, known solutions to be found during the search process. Both components are also referred to as intensification and diversification in another terminology [22]. However, these refer to medium- to long-term strategies based on the usage of memory, while exploration and exploitation refer to short-term strategies tied with randomness [5].

Essentially, randomness is used in SI algorithms in order to explore new points by moving the particles towards the search space. In line with this, several random number distributions can be helpful. For example, uniform distribution generates each point of the search space using the same probability. On the other hand, Gaussian distribution is biased towards the observed solution, this means that the smaller modifications occur more often than larger ones [9]. On the other hand, the appropriate distribution depends on the problem to be solved, more precisely, on a fitness landscape that maps each position in the search space into fitness value.

Interestingly, some authors enhanced the original FA with a chaos in order to improve it. However, a comprehensive study about chaos-based FA (CFA) are still missing. In this paper, therefore, we would like to assemble papers tackling the CFA, introduce a taxonomy of these algorithms, review the most frequently used chaotic maps in CFAs and critical discuss its advantages and disadvantages. Finally, we outline the possible directions for the future research. The purpose of this review is twofold: Firstly, to show that enhancing the CFA with chaotic maps may improve the original FA significantly, and secondly, to encourage the potential developers to start using the CFA for solving the hard problems in practice.

The structure of the paper is as follows. Section 2 describes the FA basics. In Section 3, the possible ways how to enhance the original FA with chaotic maps are explained. A brief review of the chaos-based FAs is presented in Section 4. The paper concludes with Section 5 where also the possible directions for further development are outlined.

2. The firefly algorithm

After watching the environment and sky during the summer nights, professor Yang [41] got an idea for developing a new algorithm. His source of inspiration were small lighting bugs called fireflies [6]. A phenomenon presents their flashing lights which can be seen and adored on clear summer nights. The production of these lights is done by a complicated set of chemical reactions. They flashes in order to attract a mating partner or for protection against predators. The intensity of their lights I decrease when the distance r from the light source increases in terms of $I \propto r^2$. On the other hand, air absorbs the light as the distance from the source increases [16]. This firefly behavior is modeled into the Yang's firefly algorithm (FA) so that the light intensity is proportional to the fitness function of the problem to be solved.

However, because an adaptation of the natural behavior of the fireflies in an algorithm is too complex, the following idealized rules are considered by developing of the FA:

- all fireflies are unisex,
- their attractiveness is proportional to their brightness, and
- the brightness of a firefly is affected or determined by the landscape of the objective function.

A pseudo-code of the FA is illustrated in Algorithm 1, from which it can be seen that the algorithm consists of the following elements:

- a representation of a firefly,
- an initialization (line 1 in Algorithm 1),
- a moving operator (line 8 in Algorithm 1),
- an objective function (lines 2, 11 in Algorithm 1).

Note that this SI-based algorithm does not support an explicit selection. This means, a moving of the whole population of solution depends on the position of the global best solution as well as positions of the local better (brighter) solutions in the neighborhood (line 13 in Algorithm 1). Additionally, a termination condition (line 4 in Algorithm 1) is needed in order to provide the proper termination of the FA.

Algorithm 1. Pseudo code of the basic firefly algorithm

Input: Population of fireflies $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$, objective function $f(\mathbf{x}_i)$.
Output: The best solution $\mathbf{x}_{best}$ and its value $f_{min} = \min(f(\mathbf{x}_{best}))$.

```
generate_initial_population x^(0) = (x_1^(0), ..., x_N^(0));
f(x_i^(0)) = evaluate_new_solution_and_update_light_intensity;
t = 0;
while t < MAX_GEN do
   for i = 1 to N do
      for j = 1 to N do
         if I_j > I_j then
            move_firefly_i_towards_j_using_uniform_distribution;
         end if
      end for
      f(x_i^(t)) = evaluate_new_solution_and_update_light_intensity;
   end for
   rank_fireflies_and_find_the_best;
   t = t + 1
end while
```

The FA is a population-based algorithm, where each population member (i.e., firefly) represents a candidate solution of the problem to be solved and thus denotes a point in the search space. The candidate solution $\mathbf{x}_i$ is represented as

$$\mathbf{x}_i = (x_{i1}, \ldots, x_{iD}), \quad \text{for } i = 1, \ldots, N, \tag{1}$$

where N denotes the population size, and D is the dimensionality of the problem.

An initialization of the candidate solution is performed according to the equation

$$x_{ij}^{(0)} = U(0,1) \cdot (ub_j - lb_j) + lb_j, \quad \text{for } i = 1, \ldots, N, \tag{2}$$

where $U(0,1)$ denotes a random number drawn from a uniform distribution in the interval $[0,1]$, lb_j and ub_j are the lower and the upper limits of the corresponding jth problem variable.

The variation operator bases on the lights intensity relation $I \propto r^2$ that can be expressed in mathematical form as follows

$$I(r) = I_0 e^{-\gamma r^2}, \tag{3}$$

where I_0 denotes the light intensity at the source, and γ is a fixed light absorption coefficient. In nature, two fireflies are attracted between each other in order to find an appropriate mating partner (usually the brightest). Similar to light intensity, the attractiveness β also depends on the distance r and it is calculated according to the following generalized equation

$$\beta(r) = \beta_0 e^{-\gamma r^2}, \quad \text{for } k \geqslant 1, \tag{4}$$

where β_0 denotes an attractiveness at $r = 0$.

The distance r_{ij} between two fireflies $\mathbf{x}_i$ and $\mathbf{x}_j$ is expressed as an Euclidian distance

$$r_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\| = \sqrt{\sum_{k=1}^{D} x_{ik} - x_{jk}}, \tag{5}$$

where x_{ik} and x_{jk} are the kth element of the ith and jth firefly positions within the search-space, and D denotes the dimensionality of a problem. Each firefly i moves to another more attractive firefly j, as follows

$$\mathbf{x}_i = \mathbf{x}_i + \beta_0 e^{-\gamma r_{ij}^2} (\mathbf{x}_j - \mathbf{x}_i) + \alpha \cdot \epsilon_i, \tag{6}$$

where α represents a step size scaling factor and ϵ_i is a randomization parameter. Eq. (6) consists of three terms. The first term determines the position of the ith firefly. The second term refers to a social component of moving the firefly i towards the more attractive firefly j, while the third term is connected with the randomized move of the ith firefly within the search-space.

The step size scaling factor is proportional to the characteristics scale of the problem of interest, while originally the randomization parameters denotes a random number drawn from a uniform distribution. However, the last term can also be expressed differently. For instance, Yang in [42] replaced the uniform distribution with a Lévy flights as follows

$$L(c,\lambda) = \frac{\lambda \cdot \Gamma(\lambda) \cdot \sin(\pi\lambda/2)}{\pi} \cdot \frac{1}{c^{1+\lambda}}, \tag{7}$$

where $\Gamma(\lambda)$ represents a Gamma function, $c \in \mathbb{R}$ is a scale and $\lambda \in [0,2]$ an exponent of the density function, which reduces to Cauchy distribution, when $\alpha = 1$ and to Gaussian distribution, when $\alpha = 2$.

During the evaluation phase, a value of objective function is calculated for each candidate solution. Here, the light intensity is proportional to the objective function of the problem being optimized and vice verse (i.e., $I(\mathbf{x}) \propto f(\mathbf{x})$). This means, the light intensity is updated proportionally to the value of the objective function.

Interestingly, the FA distinguishes two limiting behavior that are controlled by a setting of the parameter γ. When $\gamma \to 0$, the attractiveness in Eq. (6) becomes $\beta = \beta_0$, i.e., a firefly can be seen by all the other fireflies in the population. In this case, the FA behaves similar to the particle swarm optimization (PSO). On the other hand, when $\gamma \to \infty$, the second term is eliminated from Eq. (6), i.e., fireflies do not see each other. In this case, the firefly movement becomes similar to random search. In practice, the FA behavior is modeled using this parameter regarding to the characteristics of the problem to be solved.

3. Chaos-based firefly algorithms

Chaos in natural sciences denotes a deterministic system that behaves unpredictable [11]. In mathematics, chaos does not indicate the system with a complete absence of order, but perfectly ordered system that includes some flavor of randomness. This term was firstly used by Li and Yorke in [27] at 1975. Usually, this phenomenon is considered as a part of dynamical systems that change over the time.

Chaotic behavior is primarily detected in mathematics by iterated functions that returns the random values in each iteration. The generated sequence of values by chaotic functions (also an orbit) variates when started from the different initial value. Interestingly, each orbit limits to the same limit value, when the number of iterations goes to the infinity. In fact, such behavior is a characteristics of the ergodic systems. The orbit is usually represented as a so called chaotic maps, where the set of input values is mapped to a set of output values.

In general, characteristics of chaotic maps are determined by the following propositions [11]:

- the dynamic rule of generating the sequence of numbers is deterministic,
- the orbits are aperiodic (they never repeat),
- the orbits are bounded (time series stay between upper and lower limits, normally, within the interval $[0,1]$),
- the sequence has sensitive dependence on the initial condition (also SDIC).

Randomness is SI-based algorithms plays the primary role in the SI-based search process by exploring the new solutions. Therefore, many efforts were invested in a research of the new randomized methods. Recently, a theory of chaos offers a lot of deterministic time series of different characteristics. The chaos-based methods can successfully replace the existing random generators in many applications because of increasing the exploration power of the stochastic search process. On the other hand, these methods have a different characteristics and therefore can easier be adapted to the fitness landscapes of the different problems to be solved.

The original FA can be enhanced with chaos in two way. On the one hand, the chaotic map replaces some random distributed FA parameter in order to improve the performance [20,14]. On the other hand, the firefly intrinsic structure is applied for tuning algorithm parameters using chaotic map [46].

Three parameters control a behavior of the original FAs, i.e., step size of randomized move α, attractiveness β and absorption coefficient γ. The first parameter affects the randomized term $\alpha\epsilon_i$ in Eq. (6), where the randomized parameter ϵ_i is expressed using chaos time series, as follows

$$\epsilon_i = \mathrm{C}_i^{(k)}, \tag{8}$$

while the Eq. (4) becomes the following chaotic-enhanced form

$$\beta_i = \beta_0 \mathrm{C}_i^{(k)}, \tag{9}$$

where $\mathrm{C}_i^{(k)}$ is the particular chaotic map determined by the number k.

Interestingly, the firefly social component of move in Eq. (6) can be expressed by some simplifications according to Yang [46] as follows

$$y_i^{(t+1)} = y_i^{(t)} - \beta \exp^{-\gamma y^{(t)2}} \cdot y_i^{(t)}. \tag{10}$$

When it is assumed $u_i^{(t)} = \sqrt{\gamma} \cdot y_i^{(t)}$, the above equation can be further simplified to

$$u_i^{(t+1)} = \lambda u_i^{(t)} \left(1 - u_i^{(t)}\right). \tag{11}$$

As a result, the obtained equation prescribes a logistic chaotic map. This means that the logistic map is yet intrinsic characteristics of the social component in the FA.

Table 1
Recent applications using CFA.

CFA Variant	Chaos-enhanced parameters	Optimization	Authors	References
Modified	β, γ	Global	Yang	[45]
Modified	α, γ	Engineering	Coelho et al.	[8]
Modified	α, γ	Engineering	Coelho et Mariani	[4]
Modified	α, β	Engineering	Arul et al.	[2]
Modified	β, γ	Global	Gandomy et al.	[20]
Hybrid	α, β, γ	Engineering	Kazema et al.	[26]
Modified	β, γ	Engineering	Abdel-Raouf et al.	[1]
Modified	α	Global	Fister et al.	[14]
Modified	α, γ	Engineering	Mishra et al.	[33]
Hybrid	β, γ	Engineering	Long et Meesad	[29]
Modified	β, γ	Engineering	Yang	[46]

4. A brief review of chaotic-based firefly algorithms

An aim of this article is to review the recently work tackling the CFA. In line with this, the existing articles from this domain were assembled. Then, a detailed analysis of these articles was performed. As a result, a taxonomy of CFA was proposed. The section is followed by a detailed review of the chaotic time series methods as found in the articles. The section is finished with a discussion, where advantages and disadvantages of using the chaos in FAs are assessed and directions for the future work are indicated.

4.1. Analysis of chaos-based firefly algorithms

Papers tackling the CFA addressed in our analysis are presented in Table 1. The table consists of five columns that denote: the chaos-based FA variant, parameters that are enhanced with chaos, optimization domain to which this belong, authors of the particular paper and a reference on this. In the remainder of the paper, a detailed analysis of the mentioned papers is performed.

The first try to develop a CFA was performed by Yang [45] in 2011, which introduced a logistic map for attractiveness and absorption coefficient in place of Gaussian or Lévy flight distributed random variables. This algorithm was applied to a global optimization problem (i.e., function optimization).

Coelho et al. in [8] developed the CFA for solving the reliability–redundancy allocation that belongs to a class of reliability problems. Here, the logistic chaotic map enhanced an absorption coefficient as well as randomized parameter. This modified FA outperformed other optimization techniques, like dynamic programming, integer programming, mixed-integer programming as well as the original FA by reliability–redundancy designing of an over-speed protection system for a gas turbine. A paper [4] of Coelho et Mariani presents a continuation of the previous work, where tuning parameters of the multi-loop proportional-integral-derivative (PID) controller was used as a test bed for testing the CFA. In this case, a Tinkerbell chaotic map was incorporated in the CFA. The promising results of the proposed CFA were compared with GA, PSO, original FA and modified FA.

Arul et al. in [2] proposed a CFA for solving the economic load dispatch problem (ELD). In this paper, the randomizing and attractiveness parameters were enhanced with tent chaotic map. The reported results of the proposed algorithm were shown a good convergence characteristics on all considered ELD test cases when compared with the other soft computing techniques reported in literature.

Gandomi et al. in [20] enhanced an attractiveness and absorption coefficient in their CFA with 12 different chaotic maps and applied it to the global optimization problem. Thus, it was showed that some chaotic maps clearly outperformed the results of the original FA.

A paper of Kazema et al. [26] proposed a meta-heuristic CFA for searching the optimal values of support vector regression (SVR) parameters used to predict stock market price. In this SVR-CFA, all three parameters (i.e., randomized parameter, attractiveness and absorption coefficient) were enhanced with logistic chaotic map. The results of the proposed algorithm according the mean square error (MSE) and mean absolute percent error (MAPE) outperformed the results obtained by chaotic genetic algorithm-based SVR (SVR-GA), firefly based SVR (SVR-FA), artificial neural networks (ANN) and adaptive neuro-fuzzy inference systems (ANFIS).

Abdel et al. in [1] proposed the CFA for a parallel calculating numerical integration in engineering. This modified algorithm enhanced parameters of social component of FA move (i.e., attractiveness and absorption coefficient) with a sinusoidal chaotic map. The results of numerical simulations showed a high convergence rate, high accuracy and robustness of the proposed CFA.

Fister et al. in [14] developed the randomized FA (RFA) algorithm with various probability distributions (e.g., uniform, Gaussian and Lévy flights) as well as logistic and Kent chaotic maps. Thus, the randomized parameter was enhanced with mentioned probability distributions and chaotic maps. The results of comparison study confirmed a successfulness of incorporating the chaotic maps into the FA.

In a paper of Long and Messad [29], a hybrid CFA + GA employed as a meta-heuristic for determining the structure of fuzzy rule as well as the number of rules for fuzzy logic system that was used to predict sea water level. Indeed, the attractiveness and absorption coefficient were enhanced with Gauss chaotic map. The results obtained by this hybrid algorithm outperformed the results of both GA and FA.

The RSA public-key crypto-system problem was solved by Mishra et al. in [33] using the multithreaded bound varying CFA, where the randomized parameter and absorption coefficient were enhanced with logistic chaotic map. The results of experiments showed that this CFA is an effective tool by solving this problem.

Yang in [46] performed a comparative study covered the original FA, chaotic-enhanced FA (CFA+) and automatic parameter tuning FA (AutoFA). In the CFA+, the intrinsic structure of the FA was employed for tuning the attractiveness and absorption coefficient, while in the AutoFA, all three FA parameters were self-adapted during the SI-based search process run. The results of mentioned algorithms solving design optimization problems from literature showed that CFA+ can improve its search efficiency significantly.

4.2. Taxonomy of chaos-based firefly algorithms

In this subsection, a classification of the CFAs is identified. This classification bases on the analysis of papers illustrated in Table 1. In general, the chaos-based FAs can be divided into two classes (Fig. 1: modified CFA and hybrid CFA. The former class covers algorithms that enhanced the algorithms parameters with chaotic maps, while the latter incorporates a problem-specific knowledge into CFA structures by solving the particular problem.

The modified CFAs have been applied to theoretical or real-world problems. The hybridization of CFAs can be divided into two classes: an automatic parameter tuning and hybridization of components. The automatic parameter tuning means that the CFA acts as a higher-level heuristic that searches for an optimal parameter setting of a lower-level heuristic solving the problem of interest. The hybridization of components refers to a basic heuristic algorithm that borrows a specific components from the other heuristic algorithms. For instance, Fister et al. in [13] hybridized the Bat algorithm [44] (BA) with "rand/1/bin" mutation strategy borrowed from differential evolution [37] (DE) in order to improve the original algorithm.

The taxonomy of papers tackling the CFA is presented in Table 2, from which it can be seen that the most papers cover a solving the real-world problems.

Interestingly, the paper [29] in Table 2 was classified in both hybrid classes because the CFGA algorithm illustrated in this paper bears characteristics of the modification as well as hybridization of components because this algorithm borrowed the crossover operator from the original GA.

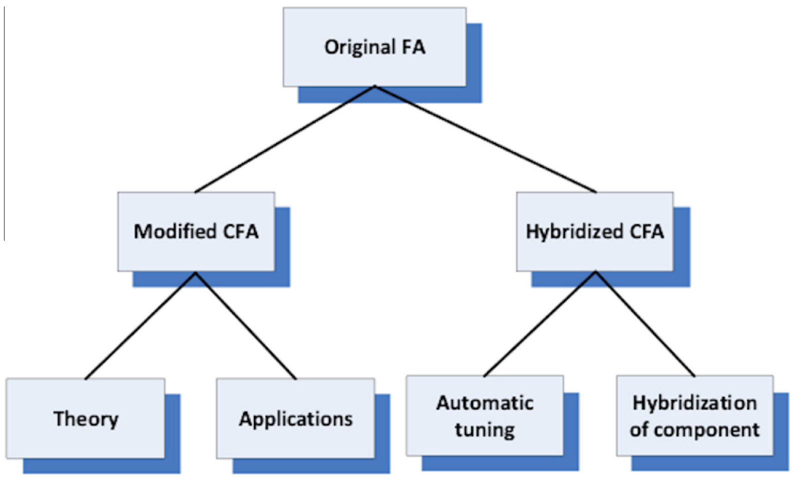

Fig. 1. An overview of the taxonomy of chaos-based firefly algorithms.

Table 2
Publications tackling the chaos-based firefly algorithms.

Topic	References
Original firefly algorithm (FA)	[41]
Theoretical chaos-based firefly algorithm (CFA)	[45,20,14]
Real-world chaos-based firefly algorithm (CFA)	[8,4,2,1,33]
Meta-heuristical chaos-based firefly algorithm (CFA)	[26,29]
Hybrid chaos-based firefly algorithm (CFA)	[29]

Table 3
chaotic maps in firefly algorithm.

Number	Chaotic maps	References
1	Chebyshev map	[20]
2	Circle map	[20]
3	Gauss map	[29,20]
4	Intermittency map	[20]
5	Iterative map	[20]
6	Kent map	[14]
7	Liebovitch map	[20]
8	Logistic map	[45,8,14,26,20,33]
9	Piecewise map	[20]
10	Sine map	[20]
11	Singer map	[20]
12	Sinusoidal map	[1,20]
13	Tent map	[2,20]
14	Tinkerbell map	[4]

4.3. Firefly algorithms and chaotic maps

Recently, a lot of chaotic maps were discovered primarily by mathematicians and physicians applicable to different domains of human activity. In line with this, the majority of these were applied to different algorithms for solving the various real-world problems. Therefore, the more applicable chaotic maps tackling the CFA are reviewed in this subsection as found in the assembled literature from Table 1. These chaotic maps are presented in Table 3 that is divided in three columns, i.e., the sequence number of the chaotic map, its name and the reference of the paper, where this was used.

From Table 3, it can be seen that the logistic map was enhanced parameters of the CFA the most frequently. More often also chaotic maps, like Gauss, sinusoidal and Tent maps were employed. Interestingly, the paper of Gandomi et al. [20] has dealt with the largest number of different chaotic maps. In the remained of the paper, characteristics of the most frequently chaotic maps used in the CFA were reviewed.

4.4. Chebyshev map

Chebyshev map is determined by the following iteration function [38]:

$$x_{n+1} = \cos(n \cos^{-1}(x_n)), \tag{12}$$

where n is an iteration number. Using this map, the chaotic time series $x_n \in [0, 1]$ are obtained.

4.5. Circle map

An iteration function representing the Circle map [23] is as follows:

$$x_{n+1} = x_n + b - (a - 2\pi) \sin(2\pi x_n) \mod (1), \tag{13}$$

where a chaotic time series $x_n \in [0, 1]$ are generated by $a = 0.5$ and $b = 0.2$.

4.6. Gauss map

Gauss map is a nonlinear iterated function given by Gaussian function [21]:

$$x_{n+1} = \exp -\alpha x_n^2 + \beta, \tag{14}$$

where the deterministic chaotic time series are produced in the interval $x_n \in [0, 1]$ for $\alpha = 4.9$ and $\beta = -0.58$.

4.7. Intermittency map

Intermittency map [10] consists of a linear and nonlinear terms, as follows:

$$x_{n+1} = \begin{cases} \epsilon + x_n + c x_n^m, & 0 < x_n \leqslant d, \\ \frac{x_n - d}{1 - d}, & d < x_n < 1, \end{cases} \tag{15}$$

where $c = \frac{1-\epsilon-d}{d^m}$, $\epsilon \ll d$, and m is usually set as $m = 2$. In line with this, the time series $x_n \in [0, 1]$ are got.

4.8. Iterative map

Iterative map [31] is introduced by the following iterative function:

$$x_{n+1} = \sin\left(\frac{a\pi}{x_n}\right), \tag{16}$$

where the chaotic time series $x_n \in [0,1]$ are obtained by setting $a \in [0,1]$.

4.9. Kent map

Kent map [11] is one of the more studied chaotic maps that has been used to generate pseudo-random numbers in many applications, like secure encryption. It is defined as follows:

$$x(n+1) = \begin{cases} \frac{x(n)}{m}, & 0 < x(n) \leqslant m, \\ \frac{(1-x(n))}{1-m}, & m < x(n) < 1, \end{cases} \tag{17}$$

where $0 < m < 1$. Hence, if $x(0) \in [0,1]$, for all $n \geqslant 1,\ x(n) \in [0,1]$.

4.10. Liebovich map

The iterative function generating the Liebovich map [38] is the following form:

$$x_{n+1} = \begin{cases} \alpha_1 x_n, & 0 < x_n \leqslant d_1, \\ \frac{p-x_n}{d_2-d-1}, & d_1 < x_n \leqslant d_2, \\ 1-\alpha_2(1-x_n), & d_2 < x_n \leqslant 1, \end{cases} \tag{18}$$

where d_i for $i = \{1,2\}$ are endpoints of the subintervals and α_i for $i = \{1,2\}$ denote the slopes of the linear maps. Thus, the chaotic time series $x_n \in [0,1]$ are generated.

4.11. Logistic map

The following iterated function define the logistic map [11]:

$$x_{n+1} = r x_n(1-x_n), \tag{19}$$

where $x_n \in [0,1]$ and r is a parameter. A generated time series $x_n \in [0,1]$ are chaotic when the iterated Logistic map with $r = 4$ is used.

4.12. Piecewise map

Piecewise map [39] consists of four linear pieces that are determined by the following iterated function:

$$x_{n+1} = \begin{cases} \frac{x_n}{d}, & 0 \leqslant x_n < d, \\ \frac{x_n-d}{0.5-d}, & d \leqslant x_n < \frac{1}{2}, \\ \frac{1-d-x_n}{0.5-d}, & \frac{1}{2} \leqslant x_n < 1-d, \\ \frac{1-x_n}{d}, & 1-d \leqslant x_n < 1, \end{cases} \tag{20}$$

where d denotes endpoints of four subintervals and can be set as $d \in [0,0,5]$. However, the generated chaotic time series capture the interval $x_n \in [0,1]$.

4.13. Sine map

The sine map [7] is based on the sine iteration function and it is defined as follows:

$$x_{k+1} = a\sin(\pi x_k), \tag{21}$$

where $0 \leqslant r \leqslant 1,\ 0 \leqslant x_n \leqslant 1$, while the time series of form $x_n \in [0,1]$ are generated by this map.

4.14. Singer map

A basis of the Singer map [35] represents the Singer iterated function. In other words, this chaotic map is defined as:

$$x_{n+1} = \mu(7.86x_n - 23.31x_n^2 + 28.75x_n^3 - 13.3x_n^4), \tag{22}$$

where the parameter μ can be captured from the interval $[0.9, 1.08]$. However, the chaotic time series are obtained in interval $x_n \in [0, 1]$.

4.15. Sinusoidal map

The sinusoidal map [28] generates a chaotic time series according to the equation:

$$x_{n+1} = ax_n^2 \sin(\pi x_n). \tag{23}$$

Note that the above equation becomes simplified form:

$$x_{n+1} = \sin(\pi x_n), \tag{24}$$

for $a = 2.3$ and $x_0 = 0.7$. The generated chaotic time series are in interval $x_n \in [0, 1]$.

4.16. Tent map

The tent map [34] is introduced by the following iterated function:

$$x_{n+1} = \begin{cases} \mu x_n, & \text{for } x_n < \frac{1}{2}, \\ \mu(1 - x_n), & \text{for } x_n \geqslant \frac{1}{2}, \end{cases} \tag{25}$$

where μ is a positive real constant. For $\mu \in [0, 2]$, this function generates a chaotic time series in interval $x_n \in [0, 1]$. Interestingly, the function transform to a logic map with $r = 4$ by $\mu = 2$.

4.17. Tinkerbell map

The Tinkerbell map [32] is a discrete map given by the following equations:

$$x_{n+1} = x_n^2 y_n^2 + a \cdot x_n + b \cdot y_n, \tag{26}$$

$$y_{n+1} = 2x_n y_n + c \cdot x_n + d \cdot y_n, \tag{27}$$

where $a, \; b, \; c, \; d$ are non-zero parameters and n is the iteration. The chaotic time series $x_n \in [0, 1]$ can be obtained for parameter values $a = 0.9, \; b = 0.6013, \; c = 2.0$, and $d = 0.5$.

4.18. Discussion

From the assembled literature tackling the CFA, it can be conducted that enhancing the chaotic maps in the CFA structures has the following advantages:

- higher convergence rate,
- higher accuracy, and
- higher robustness.

This means that the chaotic maps introduce an additional explorative power in the SI-based search process that prevents from to stuck in local optima. On the other hand, Fister et al. in [14] showed that the efficiency of used chaotic maps depends on the problem of interest. Moreover, their later study [15], where the same chaotic maps were applied to the Bat algorithm [44] showed that the efficiency also depends of the algorithm into which the specific chaotic map has been incorporated.

Applying the chaotic maps in place of the random probability distributions has also some technical advantages, i.e., the random generator on digital computers generates an integer variable of length 32-bit that can represents the maximal $2^{32} - 1$ $(6.5 \cdot 10^4)$ different random values. Consequently, the chaotic map time series return values from the interval $[0.0, 1.0]$ of type double float. In this case, the generated 64-bit values comprise range between 2^{52} and 2^{53}, i.e., $4.5 \cdot 10^{15}$ different random values. This fact is important for all stochastic meta-heuristic algorithms because an additional diversity of population is obtained using these chaotic maps.

Unfortunately, using of chaotic maps has also some disadvantages. The primary issue is the enormous number of chaotic maps that prevent a selection of the most promising chaotic map for some specific problem. As a possible solution of this problem, an adapted CFA could be developed in the future that will select the best chaotic map from an ensemble according to the problem of interest, the current search process state or the type of algorithm enhanced with a chaotic map.

5. Conclusions

This paper presents a brief review of the chaos-based FA. In line with this, papers tackling the CFAs are assembled, a taxonomy of these algorithms is introduced, the most frequently used chaotic maps are highlighted, advantages and disadvantages of these maps are exposed, and the possible directions for the future development of CFA are outlined.

The reported results of the CFAs obtained in the literature show that enhancing the CFAs with chaos improves characteristics of the original FA in sense of higher convergence rate, accuracy and robustness. On the other hand, the efficiency of the specific chaotic map enhancing the CFAs depends on the problem of interest as well as the basic algorithm enhanced with this chaotic map. In order to overcome these problems, the adapted CFA is proposed as a promising direction for the future work.

Acknowledgments

This project was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, under Grant No. (76-130-35-HiCi). The authors, therefore, acknowledge technical and financial support of KAU.

References

[1] Osama Abdel-Raouf, Mohamed Abdel-Baset, Ibrahim El-henawy, Chaotic firefly algorithm for solving definite integral, Int. J. Computer Science and Information Technologies 6 (2014) 19–24.
[2] R. Arul, S. Velusami, G. Ravi, Chaotic firefly algorithm to solve economic load dispatch problems, in: 2013 International Conference on Green Computing, Communication and Conservation of Energy (ICGCE), IEEE, 2013, pp. 458–464.
[3] Nebojsa Bacanin, Milan Tuba, Firefly algorithm for cardinality constrained mean-variance portfolio optimization problem with entropy diversity constraint, Scientific World J. 2014 (2014).
[4] Leandro dos Santos Coelho, Viviana Cocco Mariani, Firefly algorithm approach based on chaotic Tinkerbell map applied to multivariable PID controller tuning, Comput. Math. Appl. 64 (8) (2012) 2371–2382.
[5] Matej Crepinsek, Shih-Hsi Liu, Marjan Mernik, Exploration and exploitation in evolutionary algorithms: a survey, ACM Comput. Surv. 45 (3) (2013) 35.
[6] Jeffrey R. De Wet, K.V. Wood, Marlene DeLuca, Da R. Helinski, S. Subramani, Firefly luciferase gene: structure and expression in mammalian cells, Mol. Cell. Biol. 7 (2) (1987) 725–737.
[7] R.L. Devaney, An Introduction to Chaotic Dynamical Systems, Addison-Wesley, 1987.
[8] Leandro dos Santos Coelho, Diego Luis de Andrade Bernert, Viviana Cocco Mariani, A chaotic firefly algorithm applied to reliability–redundancy optimization, IEEE Congress on Evolutionary Computation (CEC), IEEE, 2011, pp. 517–521.
[9] A.E. Eiben, J.E. Smith, Introduction to Evolutionary Computing, Springer-Verlag, Berlin, 2003.
[10] A. Erramilli, R.P. Singh, P. Pruthi, Modeling Packet Traffic with Chaotic Maps, Royal Institute of Technology, Stockholm-Kista, Sweden, 1994.
[11] David P. Feldman, Chaos and Fractals: An Elementary Introduction, OUP Oxford, 2012.
[12] I. Fister, X.S. Yang, J. Brest, I. Fister Jr., Memetic self-adaptive firefly algorithm, Swarm Intell. Bio-Inspired Comput.: Theory Appl. (2013) 73–102.
[13] Iztok Fister, Simon Fong, Janez Brest, Iztok Fister, A novel hybrid self-adaptive bat algorithm, Scientific World J. 2014 (2014).
[14] Iztok Fister, Xin-She Yang, Janez Brest, Iztok Fister Jr., On the randomized firefly algorithm, in: Cuckoo Search and Firefly Algorithm, Springer, 2014, pp. 27–48.
[15] I. Fister Jr., X.-S. Yang, J. Brest, D. Fister, I. Fister, Analysis of randomization methods in swarm intelligence, Int. J. Bio-Inspired Comput. (2015). in press.
[16] Iztok Fister Jr., Xin-She Yang, Iztok Fister, Janez Brest, Memetic firefly algorithm for combinatorial optimization, arXiv preprint arXiv:1204.5165, 2012.
[17] A. Galvez, A. Iglesias, New memetic self-adaptive firefly algorithm for continuous optimization, Int. J. Bio-Inspired Comput. (2015). in press.
[18] Akemi Gálvez, Andrés Iglesias, Firefly algorithm for polynomial bézier surface parameterization, J. Appl. Math. 2013 (2013).
[19] Akemi Gálvez, Andrés Iglesias, From nonlinear optimization to convex optimization through firefly algorithm and indirect approach with applications to cad/cam, Scientific World J. 2013 (2013).
[20] A.H. Gandomi, X.-S. Yang, S. Talatahari, A.H. Alavi, Firefly algorithm with chaos, Commun. Nonlinear Sci. Numer. Simul. 18 (1) (2013) 89–98.
[21] D. He, C. He, L. Jiang, H. Zhu, G. Hu, Chaotic characteristic of a one-dimensional iterative map with infinite collapses, IEEE Trans. Circuits. Syst. 48 (2001) 900–906.
[22] A. Hertz, D. Taillard, E. end de Werra, Tabu search, in: E. Aarts, J.K. Lenstra (Eds.), Local Search in Combinatorial Optimization, Princeton University Press, New Jersey, 2003, pp. 121–136.
[23] R.C. Hilborn, Chaos and Nonlinear Dynamics: An Introduction for Scientists and Engineers, Oxford University Press, New York, 2004.
[24] H.H. Hoos, T. Stützle, Stochastic Local Search: Foundations and Applications, Morgan Kaufmann, 2004.
[25] Ming-Huwi Horng, Vector quantization using the firefly algorithm for image compression, Expert Syst. Appl. 39 (1) (2012) 1078–1091.
[26] Ahmad Kazem, Ebrahim Sharifi, Farookh Khadeer Hussain, Morteza Saberi, Omar Khadeer Hussain, Support vector regression with chaos-based firefly algorithm for stock market price forecasting, Appl. Soft Comput. 13 (2) (2013) 947–958.
[27] Tien-Yien Li, James A. Yorke, Period three implies chaos, Am. Math. Mon. 82 (10) (1975) 985–992.
[28] Y. Li, S. Deng, D. Xiao, A novel hash algorithm construction based on chaotic neural network, Neural Comput. Appl. 20 (2011) 133–141.
[29] Nguyen Cong Long, Phayung Meesad, An optimal design for type-2 fuzzy logic system using hybrid of chaos firefly algorithm and genetic algorithm and its application to sea level prediction, J. Intell. Fuzzy Syst. 27 (3) (2014) 1335–1346.
[30] Szymon Łukasik, Sławomir Żak, Firefly algorithm for continuous constrained optimization tasks, Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems, Springer, 2009, pp. 97–106.
[31] R.M. May, Simple mathematical models with very complicated dynamics, Nature 261 (1976) 459–467.
[32] P.E. McSharry, P.R.C. Ruffino, Asymptotic angular stability in non-linear systems: rotation numbers and winding numbers, Dyn. Syst. 18 (2003) 191–200.
[33] Mohit Mishra, Utkarsh Chaturvedi, Saibal K. Pal, A multithreaded bound varying chaotic firefly algorithm for prime factorization, 2014 IEEE International Advance Computing Conference (IACC), IEEE, 2014, pp. 1322–1325.
[34] E. Ott, Chaos in Dynamical Systems, Cambridge University Press, Cambridge, 2002.
[35] H. Peitgen, H. Jurgens, D. Saupe, Chaos and Fractals, Springer-Verlag, Berlin, 1992.
[36] J. Senthilnath, S.N. Omkar, V. Mani, Clustering using firefly algorithm: performance study, Swarm Evol. Comput. 1 (3) (2011) 164–171.
[37] Rainer Storn, Kenneth Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (4) (1997) 341–359.
[38] M.S. Tavazoei, M. Haeri, Comparison of different one-dimensional maps as chaotic search pattern in chaos optimization algorithms, Appl. Math. Comput. 187 (2007) 1076–1085.
[39] A.G. Tomida, Matlab toolbox and gui for analyzing one-dimensional chaotic maps, in: International Conference on Computational Sciences and Its Applications ICCSA, IEEE Press, 2008, pp. 321–330.
[40] X.S. Yang, Firefly algorithm, Nature-Inspired Metaheuristic Algorithms, 2008. pp. 79–90.
[41] Xin-She Yang, Firefly algorithms for multimodal optimization, in: Stochastic Algorithms: Foundations and Applications, Springer, 2009, pp. 169–178.
[42] Xin-She Yang, Firefly algorithm, levy flights and global optimization, in: Research and Development in Intelligent Systems XXVI, Springer, 2010, pp. 209–218.
[43] Xin-She Yang, Firefly algorithm, stochastic test functions and design optimisation, Int. J. Bio-Inspired Comput. 2 (2) (2010) 78–84.

[44] Xin-She Yang, A new metaheuristic bat-inspired algorithm, in: Nature inspired cooperative strategies for optimization (NICSO 2010), Springer, 2010, pp. 65–74.

[45] Xin-She Yang, Chaos-enhanced firefly algorithm with automatic parameter tuning, Int. J. Swarm Intell. Res. (IJSIR) 2 (4) (2011) 1–11.

[46] Xin-She Yang, Analysis of firefly algorithms and automatic parameter tuning, in: Emerging Research on Swarm Intelligence and Algorithm Optimization, IGI Global, 2015, pp. 36–49.

[47] Xin-She Yang, Seyyed Soheil Sadat Hosseini, Amir Hossein Gandomi, Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect, Appl. Soft Comput. 12 (3) (2012) 1180–1186.